Scheduling Algorithm

จัดทำโดย
นายสุริยา เตชะลือ 600610790

เสนอ
รศ.ดร.นริศรา เอี่ยมคณิตชาติ

สารบัญ

Concept & code of 3 algorithm

First Come First Search

```
def FCFS(FCFSTime, processList, numProcess):
    temp = 0
    for x in range(len(processList)):
        FCFSTime += temp
        temp += processList[x]
    FCFSTime = FCFSTime/numProcess
    return FCFSTime
```

ทำการประมวลผลตัวแรกที่เข้ามา Process ก่อนจากนั้นเรียงลำดับถัดไป waiting time จะเพิ่มขึ้นตามจำนวนที่ process ก่อนหน้าใช้ในการรอ และ average waiting time เท่ากับ waiting time ของ process ทั้งหมด หารด้วย จำนวน process ทั้งหมด ฟังก์ชัน FCFS คืนค่า average waiting time ของ First Come First Search

Shortest Job First

```
def SJF(SJFTime, tempprocessList, numProcess):
    temp = 0
    for x in range(len(tempprocessList)):
        SJFTime += temp
        temp += min(tempprocessList)
        tempprocessList.remove(min(tempprocessList))
    SJFTime = SJFTime/numProcess
    return SJFTime
```

เนื่องจากไม่มีการคิด arrival time ดังนั้น process ทุกตัวจึงมีลำดับการประมวลที่เท่ากัน เพราะฉะนั้นจึงให้ process ที่มีเวลาน้อยที่สุดมาประมวลผลก่อน จึงดึงค่าที่น้อยที่สุดออกจาก list process เพื่อมาประมวลผล waiting time และ average waiting time เท่ากับ waiting time ของ process ทั้งหมด หารด้วย จำนวน process ทั้งหมด ฟังก์ชัน SJF คืนค่า average waiting time ของ Shortest Job First

Round Robin

```
def RR(RRTime, temp2processList, numProcess):
    temp = 0
    point = 0
    timeQuantum = (max(temp2processList)+min(temp2processList))//2
    while len(temp2processList) > 0:
        RRTime += temp
        point = point % (len(temp2processList))
        if temp2processList[point] >= timeQuantum:
            temp += timeQuantum
            temp2processList[point] -= timeQuantum
        elif temp2processList[point] < timeQuantum:
            temp += temp2processList[point]
            temp2processList.remove(temp2processList[point])
            point -= 1
        point += 1
    RRTime = RRTime/numProcess
    return RRTime
```

การจัดคิวให้แต่ละ process เข้าไปประมวลผลตามลำดับ แต่มี Time Quantum ในการกำหนดเวลาที่แต่ละ process เข้าไปทำงานได้ โดยถ้าหมด Time Quantum ก็จะไปทำยัง process ถัดไป ค่าที่ได้จากการเรียกใช้ฟังก์ชันคือ average waiting time ของ Round Robin

Experimental results

Data set

Number of Process : 10

number of process	case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8	case 9	case 10
2 - 8 ms	23	22	22	24	13	21	23	17	21	34
20 - 30 ms	19	7	6	14	7	9	16	9	6	13
35 - 40 ms	9	7	3	4	2	3	7	6	5	6
total	51	36	31	42	22	33	46	32	32	53

percent of process	case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8	case 9	case 10
2 - 8 ms	45.09804	61.11111	70.96774	57.14286	59.09091	63.63636	50	53.125	65.625	64.15094
20 - 30 ms	37.2549	19.44444	19.35484	33.33333	31.81818	27.27273	34.78261	28.125	18.75	24.5283
35 - 40 ms	17.64706	19.44444	9.677419	9.52381	9.090909	9.090909	15.21739	18.75	15.625	11.32075

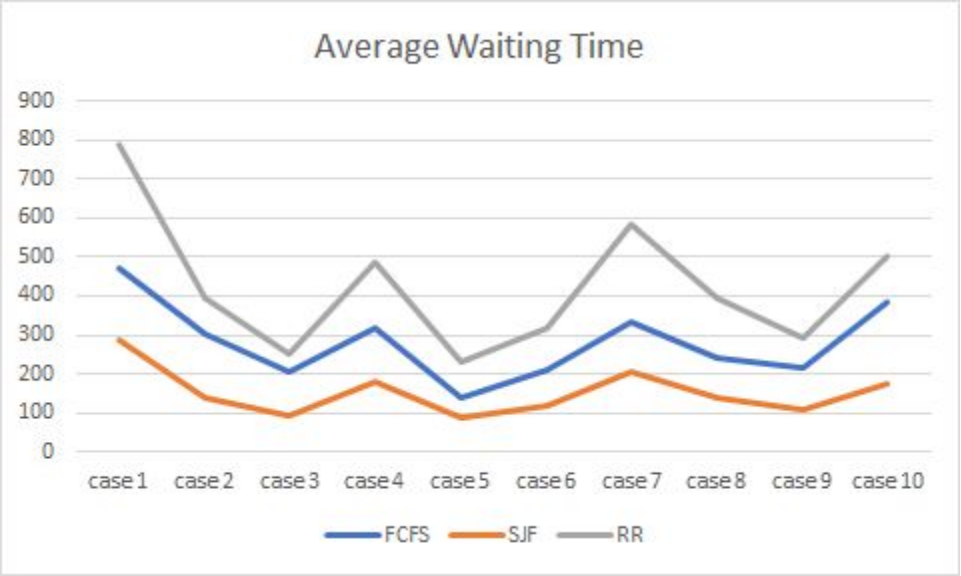
average waiting time	case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8	case 9	case 10
FCFS	472.6078	302.1389	206.871	316.9524	141.8182	213.303	335.8478	242.9688	214.4063	383.9811
SJF	285.7451	136.8056	95.06452	179.3333	87.31818	121.0303	204.1739	137.9688	108.9375	172.5094
RR	785.8235	396.1667	250.0968	485	231.1364	315.8182	582.7174	392.6563	290.5	501.3019

Data list										
No. case	case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8	case 9	case 10
Sum time process	959	546	392	647	326	451	760	519	447	713
	35	39	5	38	4	37	21	2	4	3
	7	8	38	6	8	3	26	7	29	21
	38	25	7	28	7	7	5	36	27	30
	7	5	4	8	21	7	2	4	5	3
	7	6	5	39	2	7	3	21	7	3
	8	25	8	21	6	28	2	28	4	2
	39	2	29	8	22	3	27	7	5	6
	2	29	27	3	39	8	8	38	25	37
	29	3	2	27	8	27	28	22	3	6
	28	3	8	6	21	4	4	5	7	35
	8	25	28	6	7	7	4	5	36	28
	3	35	3	6	4	21	20	3	3	2
	30	28	5	30	37	21	7	27	8	38
	4	20	36	36	6	8	28	27	2	39
	3	2	4	23	8	2	2	21	28	38
	29	36	8	4	28	7	3	3	30	25
	39	8	36	6	28	4	21	7	5	2
	6	8	2	2	27	6	6	23	4	25
	39	36	29	21	5	5	29	37	35	4
	28	8	3	5	7	2	23	6	20	4
	7	6	2	5	4	24	8	2	5	3
	28	22	29	7	27	27	21	8	37	26

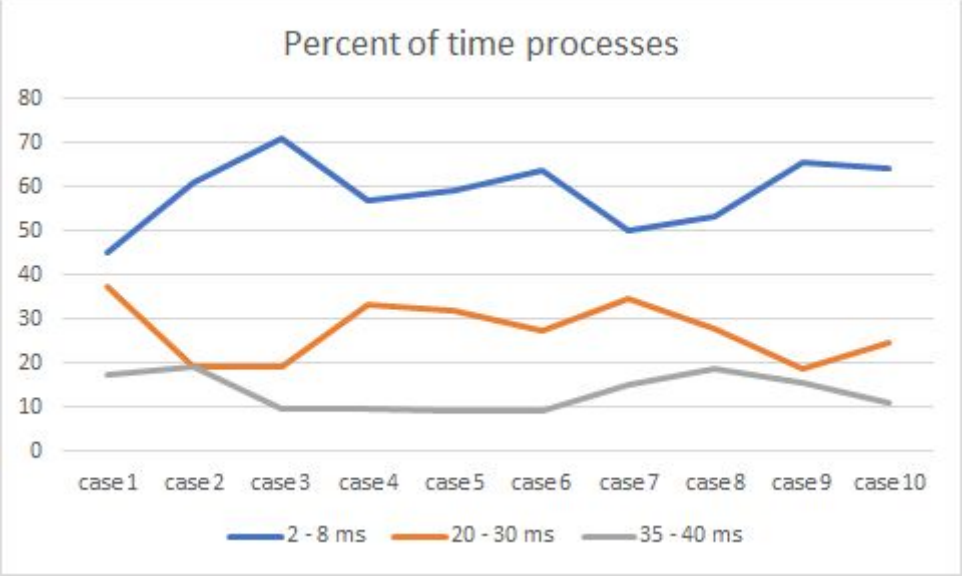
	25	35	2	8		39	4	36	8	5
	35	38	7	3		7	2	7	8	4
	5	6	7	23		36	39	3	4	2
	4	8	26	6		25	4	5	38	30
	4	4	7	22		23	39	21	6	21
	35	39	8	23		7	38	25	4	8
	6	2	4	8		23	30	5	2	7
	39	7	6	2		8	2	5	5	2
	5	8	7	6		5	21	37	38	7
	22	3		5		7	2	36	5	3
	30	8		27		6	28			27
	20	3		26			21			8
	5	2		29			38			8
	4	4		8			22			3
	25			5			4			8
	35			5			39			7
	30			8			7			5
	29			30			8			8
	22			29			36			6
	3			39			4			2
	24						24			5
	6						5			28
	6						38			24
	8						7			28
	22									37
	6									4

	29									25
	27									3
	24									2
										2
										4

Results



กราฟเปรียบเทียบ average waiting time ระหว่าง First come first search, Shortest job first และ Round robin



กราฟเปรียบเทียบอัตราส่วนของเวลาในแต่ละ process ระหว่าง First come first search,Shortest job first และRound robin

Conclusion

จากข้อมูล Data set ข้างต้นที่ได้นำมาทดลองรันผ่านฟังก์ชั่น FCFS, SJF และ RR และลัพธ์ของเวลาที่ได้ เมื่อนำมาสร้างกราฟ Average waiting time แสดงให้เห็นว่า อัลกอริทึมในการจัดเวลาให้แต่ละ Process เข้าไปทำงานยัง CPU อัลกอริทึม Shortest job first ดีที่สุดเพราะให้ process ที่ใช้เวลาในการประมวลผลน้อยที่สุดทำงานก่อน ทำให้ average waiting time น้อยที่สุด รองลงมาคือ First come first search ที่ให้ process ที่มาก่อนได้ทำงานก่อน และ อัลกอริทึมที่ใช้ในการจัดคิวการทำงานของ CPU ได้มีประสิทธิภาพน้อยที่สุดจากการทำงานนี้ก็คือ Round robin เพราะมีการกำหนด Time Quantum เพราะเมื่อแต่ละ process ยังทำงานยังไม่เสร็จแต่ Time Quantum หมดจึงต้องออกจาก CPU ทำให้ process ที่ยังทำงานไม่เสร็จมี average waiting time มากกว่า อัลกอริทึมของ First come first search และ Shortest job first